

Proposed System: Quantum Computing Processor Based on Linear Number Relation

Mayur Nilesh Barhate^{1,*}

Abstract

This paper details the design and implementation of a specialized quantum processor tailored for efficiently evaluating custom mathematical formulas. Optimized to perform a specific set of arithmetic and logical operations, this processor delivers a stable and high-performance computing platform. Unlike general-purpose quantum processors, which are designed for versatility across a wide range of algorithms, this dedicated processor aims to enhance performance and accuracy for a targeted set of tasks. The paper starts by introducing the motivation behind creating a specialized quantum processor and the benefits it provides compared to using standard logic gates. The methodology section outlines the design principles, circuit architecture, and optimization strategies used in developing the processor. It highlights the customization of gate operations and circuit design to align precisely with the formulated equations, ensuring peak performance. Simulation results highlight the effectiveness of the quantum processor in efficiently assessing custom-formulated formulas, revealing superior speed and accuracy compared to standard quantum computing methods. The discussion explores the impact of using a dedicated processor for specialized computations and emphasizes its potential for broader applications in fields that demand specific mathematical operations.

Keywords: Linear, number, relation, formula, quantum, computing, processor, logic gate

INTRODUCTION

Quantum computing signifies a transformative leap in computational capability, offering the potential for exponential speed improvements in solving complex problems that classical computers struggle with. A major challenge in quantum computing is the development of processors and algorithms that can manage specialized computations effectively [1]. This study focuses on addressing this challenge by introducing a dedicated quantum processor that is specifically optimized for evaluating linear number relation formulas.

Linear number relations are essential in mathematics and have applications across various domains such as cryptography, optimization, and scientific modeling. Traditional computing methods often struggle

*Author for Correspondence

Mayur Nilesh Barhate
E-mail: mayumb2003@gmail.com

¹Student, Department of Computer Science and Engineering (Internet of Things and Cyber Security Including Block Chain Technology), Jawahar Education Society's A.C. Patil College of Engineering, Kharghar, Navi Mumbai, Maharashtra, India

Received Date: June 28, 2024
Accepted Date: September 12, 2024
Published Date: October 08, 2024

Citation: Mayur Nilesh Barhate. Proposed System: Quantum Computing Processor Based on Linear Number Relation. Recent Trends in Parallel Computing. 2024; 11(3): 24–32p.

with the computational complexity of these formulae, particularly when dealing with large datasets or intricate mathematical relationships. Quantum computing is a promising solution that utilizes quantum superposition and entanglement to execute parallel computations, thereby overcoming the constraints of classical computing [2]. The objective of this study is to design and optimize a quantum computing processor tailored to operate seamlessly using linear number relation formulas. By customizing the gate operations, circuit architecture, and optimization strategies, the processor aims to achieve enhanced performance, accuracy, and stability, specifically for these computations.

In this study, we present the design principles, methodology, simulation results, and discussion related to the development of a dedicated quantum processor [3]. The results show the efficacy of the processor in efficiently evaluating linear number relation formulas, paving the way for advancements in specialized quantum computing applications.

Quantum computing is a groundbreaking field that leverages the principles of quantum mechanics to address complex calculations beyond those of traditional computers. Unlike conventional computers that use bits to represent data as 0s and 1s, quantum computers use qubits. Qubits can exist in multiple states simultaneously, owing to superposition and entanglement, enabling them to perform more advanced computations [4]. Linear numeric relationships are mathematical formulas that involve linear combinations of numbers and operations such as addition, subtraction, multiplication, and division. These relationships play an important role in various mathematical models, optimization problems, and data analysis problems.

The LNR quantum processor was specifically designed to compute linear numericities using quantum-computing principles. Using quantum gates, qubit arrays, and advanced algorithms, the processor was designed to significantly improve the efficiency and speed of estimating linear numerical relationships, paving the way for advancements in quantum computing applications.

METHODOLOGY

Problem Formulation

Linear number relation formula:

Formula 1:

$$m = \sqrt{\left(\frac{((b + b) - (b - b) - (b * b) - \left(\frac{b}{b}\right))}{(-1)}\right)}$$
$$n = \sqrt{\left(\frac{((b + b) + (b - b) + (b * b) + \left(\frac{b}{b}\right))}{(1)}\right)}$$

Formula 2:

$$b = \sqrt{\left[\frac{((a*a)+(c*c))}{2}\right] - \left[\frac{(c-a)}{2}\right]}$$

Note: 'a' and 'c' it is respective, 'm' and 'n' mean m = a and n = c.

QUANTUM COMPUTING CONCEPT

Qubits

- Quantum bits are the core elements of quantum information in quantum computing. Unlike classical bits, which are limited to either 0 or 1, qubits can exist as a superposition of states, representing both 0 and 1 simultaneously.
- Superposition enables parallel processing and the potential for exponential computational speedups in quantum algorithms.

Quantum Gates

- Quantum gates are operations applied to qubits to perform computations in quantum circuits.
- Common quantum gates include the Hadamard (H), Pauli-X (X), Pauli-Y (Y), Pauli-Z (Z), and controlled-NOT (CNOT) gates.
- These gates manipulate qubit states, create entanglement, and perform quantum transformations that are essential for quantum computations.

Superposition and Entanglement

- Superposition enables qubits to represent multiple states simultaneously, facilitating parallel processing and enhancing computational efficiency.

- Entanglement is a quantum phenomenon in which qubits become interconnected and exhibit correlations, regardless of the distance between them. This enables the creation of quantum states with unique properties, which are used in quantum algorithms.

Quantum Measurement

- Quantum measurement involves retrieving classical information from qubits by causing their superposition states to collapse [5].
- The measurement outcomes are probabilistic, following the rules of quantum mechanics, and can yield different results upon repeated measurement.

Quantum Circuit Model

- Quantum circuits are representations of quantum algorithms that are composed of qubits, quantum gates, and measurements.
- Quantum algorithms are executed using sequences of quantum gates applied to the qubits, leading to computational transformations and results.

Quantum Parallelism and Quantum Fourier Transform

- Quantum parallelism allows quantum algorithms to simultaneously evaluate multiple possibilities by leveraging superposition and entanglement [6].
- Quantum Fourier Transform (QFT) is a powerful quantum algorithm used in applications such as factoring large numbers and solving linear equations efficiently.

Architecture of the Quantum Processor

Qubit Arrangement

- The quantum processor consists of qubits arranged in a suitable configuration to perform the required computations efficiently.
- For linear number relation evaluations, a register of qubits was organized to store the input values and intermediate results during computation [7].

Ancilla Qubits

- Ancilla qubits are used for error correction, gate operations, or specific computational tasks.
- In quantum processors for linear number relations, ancilla qubits may be employed to implement certain quantum algorithms, enhance precision, or manage the quantum states.

Quantum Gates and Their Functions

Hadamard Gate (H)

- The Hadamard gate generates superposition by converting a qubit from the $|0\rangle$ state into an equal superposition of $|0\rangle$ and $|1\rangle$.
- Linear number relation computations are used to initialize the qubits in the superposition states for parallel processing.

Pauli-X Gate (X)

- The Pauli-X gate is a bit-flip gate that changes the state of a qubit by flipping it from $|0\rangle$ to $|1\rangle$ and vice versa.
- Employed in operations involving negation or inversion of qubit states in linear number-relation evaluations.

Pauli-Z Gate (Z)

- The Pauli-Z gate applies a phase shift of π radians (180 degrees) to the qubit state $|1\rangle$.
- It is used in quantum algorithms for phase manipulation and phase estimation and is potentially relevant in certain linear number relation computations.

Controlled-NOT Gate (CNOT)

- The controlled-NOT gate performs a controlled operation in which the target qubit is flipped (X-gate) if the control qubit is in the $|1\rangle$ state.
- It has been applied in quantum circuits for conditional operations and entanglement generation, facilitating complex computations in linear number relations [8].

By strategically incorporating these quantum gates and organizing qubits and ancilla qubits in an optimized architecture, a quantum processor can efficiently evaluate linear number relation formulas, leveraging quantum parallelism and entanglement to enhance computational capabilities.

According to conceptual implementation of the processor, it is as follows: Figures 1 and 2 show the logical design of LNR Formulas 1 and 2, respectively. Therefore, we know the actual implementation of the processor using the quantum logic gate, but its conceptual design paper only considers a conceptual introduction. Figures 1 and 2 show component names and their meanings.

- 0 to 9 means 0 to 9 numbers describe numbers.
- Convert B means 0 to 9 in number select then this number converts qubit.

In addition, subtraction, multiplication, and divisions are fundamental operations in our math and LNR formulas to help produce previous and next numbers. At time 4, operation means that we discuss the fundamental operation, so they produce four different outputs at a time. Produce four outputs at the time of addition and subtraction so that they can form a square.

Sqrt solves it is solved square and finds root so previous on we can find two outputs put in this then we can find at time 2 output it is previous and next number and in between number it is our selected number. M and N represent the previous discussion, according to the next number. Then the second phase process design is based on the LNR formula 2.

Therefore, previous dispersion according to M and N goes through a separate multiplication process at the same time M and N go for subtraction, then multiplication values produce two values: square and time subtraction also produces value [9].

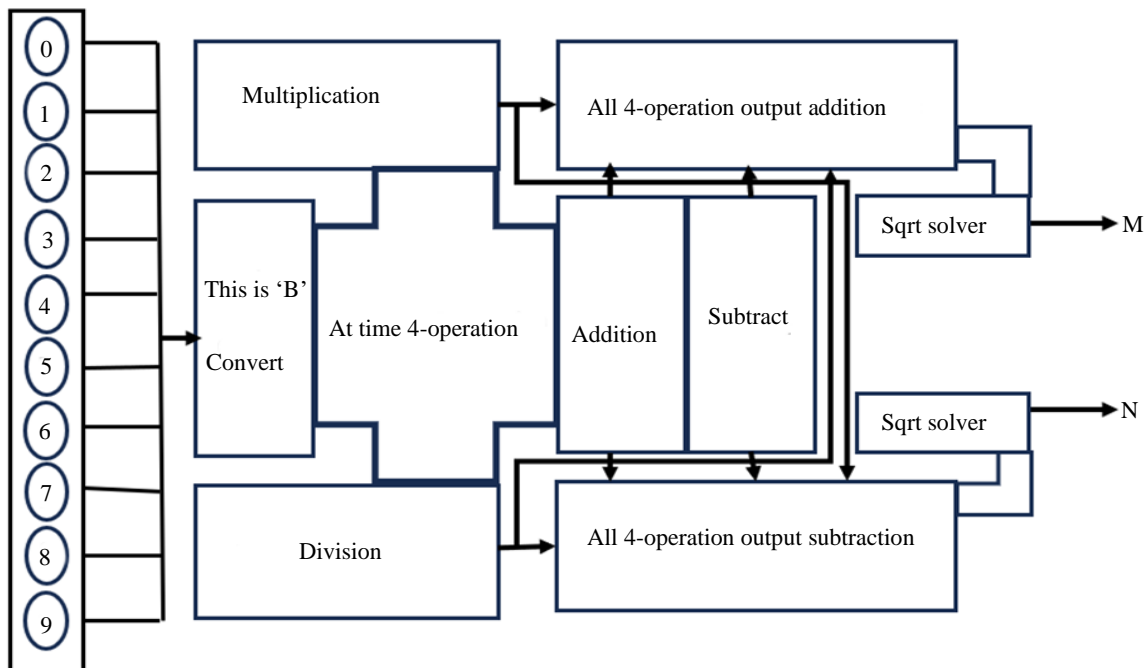


Figure 1. LNR formula 1 according to processor phase 1 design.

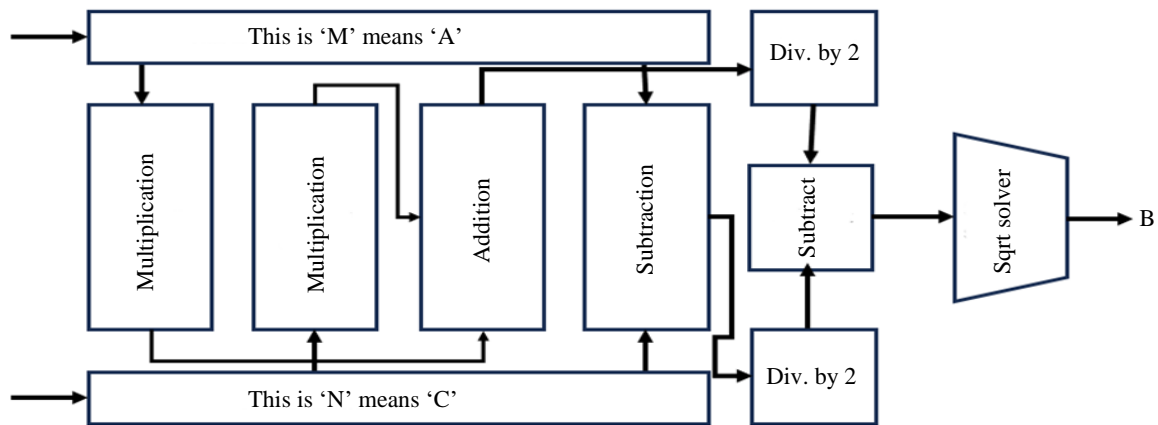


Figure 2. LNR Formula 2 According to the processor phase 2 design.

The added values produce one output, and the subtracted value is already produced, so they are divided by 2, so they produce different 2 values. Their final subtraction produces the final value. The final value placed in the square root and producing B is the initially selected number from 0 to 9, similar to the same number produced.

TESTING PROCESS

Input Data Sets

- Prepare a range of input data sets representing different linear number relation formulas.
- Varying complexities, sizes, and input parameters were included to assess the scalability and accuracy of the quantum processor.

Expected Outputs

- The expected outputs for each input dataset were determined based on classical computing results or analytical solutions.
- The criteria for correctness, precision, and performance expectations are defined to evaluate the outputs of the quantum processor.

Quantum Circuit Execution

- Implementation of quantum circuits corresponding to linear number relation formulas in the simulation environment.
- The circuits are executed using quantum gates and operations specified in the quantum processor design.

Measurement and Analysis

- Perform quantum measurements on the final quantum states to obtain measurement outcomes.
- The measurement results were analyzed and compared with the expected outputs to validate the functionality of the quantum processor.

PERFORMANCE METRICS

Accuracy

- The accuracy of a quantum processor is measured by comparing its output with that of known solutions or classical computing results.
- Error rates or deviations were calculated to quantify the level of accuracy achieved in evaluating the linear number relations.

Speed

- The computational speed of the quantum processor was evaluated by measuring the time required to execute the quantum circuits for different input datasets.

- The processing speed was compared with that of classical computing methods to assess the potential speedups achieved through quantum parallelism.

Resource Utilization

- Monitor resource utilization metrics such as memory usage, qubit coherence times, and gate fidelity during simulation and testing.
- Optimize resource allocation and quantum gate operations to minimize resource overhead and improve overall performance.

By systematically conducting simulations and testing procedures with diverse input datasets, expected outputs, and performance metrics, we can effectively evaluate the performance, accuracy, and scalability of a quantum processor for linear number relation computations [10].

VERIFICATION AND VALIDATION

The conceptual design of a quantum processor requires verification and validation during its actual implementation. The processor is not designed to handle multiple operations but is instead a dedicated processor. The following sections outline the verification and validation process.

Note: It is not considered experimental because of the high cost of quantum hardware and the high rate of hardware failure; this paper provides a conceptual design with suggestions in methodology to provide how we can make so do not assume it is implemented just conceptual think and work.

Conceptual Process

Figure 3 describes the actual execution. The process begins with a number ranging from 0 to 9. When a number is chosen, it is considered as ‘B’. The formulas mentioned below can help understand the workings of the process. First, the number is converted to binary. Then, 4 operations are performed, producing 4 values, after which the positive binary is obtained. The square root (sqrt) is solved, and the result is a pair of numbers between 0 and 9. These numbers proceed to the next step, where they are denoted as M = A and N = C. Next, multiplication operations between A and C are performed, followed by addition and subtraction operations. After dividing by 2, the result undergoes a subtraction with the square root solution. This yields ‘B’, which is the same value chosen as the input. The processor performs multiple operations to handle various values, which can enhance its features according to the needs of the process. While the implementation is complex, further advancements will provide optimized services. Table 1 verify the input/output. Figure 3 illustrates the process of the solution.

Select any number from 0 to 9; therefore, we selected 4 with a mean B = 4. Then, according to the LNR Formula value, we obtain the following equation:

Formula 1:

$$m = \sqrt{((4 + 4) - (4 - 4) - (4 * 4) - (\frac{4}{4}))/(-1)}$$

$$n = \sqrt{((4 + 4) + (4 - 4) + (4 * 4) + (\frac{4}{4}))/1}$$

$$m = 3$$

$$n = 5$$

$$m = a \text{ and } n = c$$

Table 1. Processor output verification and validation.

Any number	Previous number	Next number	Same number for choose as any number
2	1	3	2
9	8	10	9

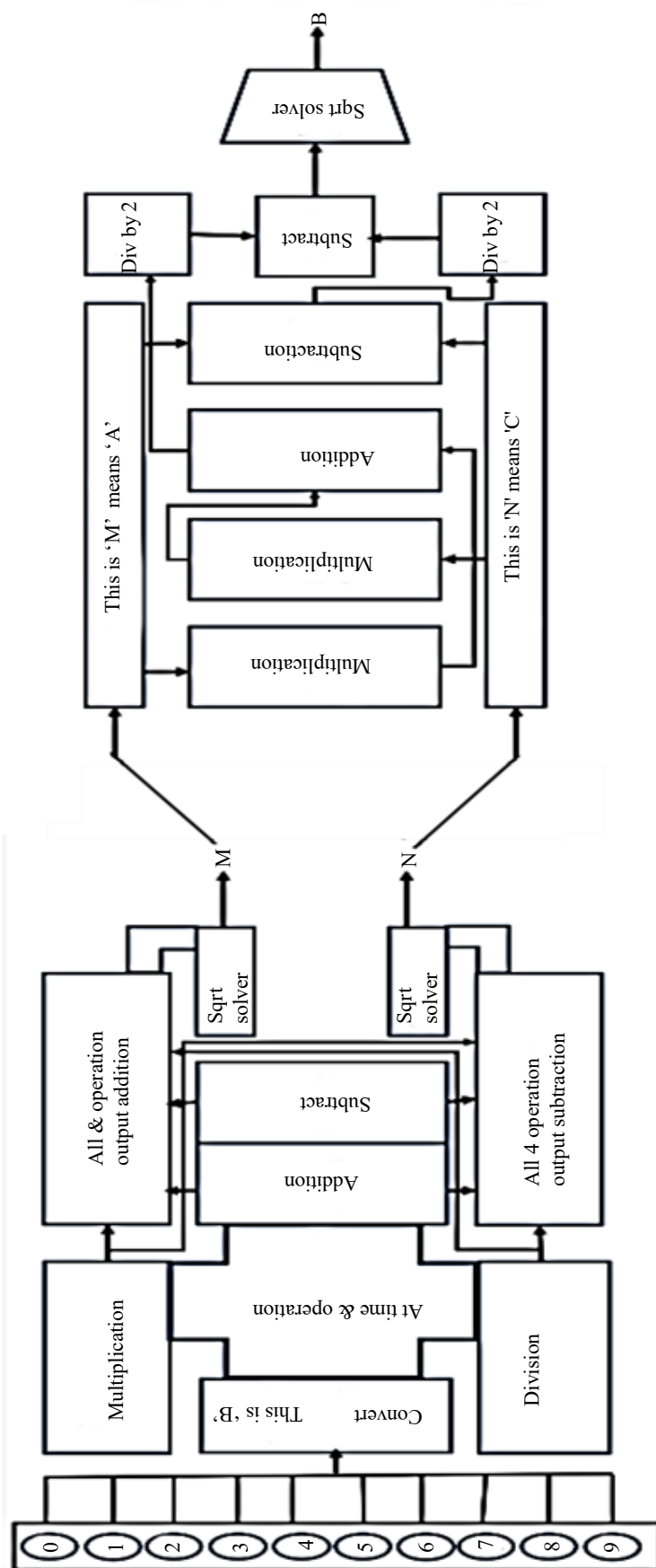


Figure 3. LNR quantum computing processor (Phases I and II combine).

Formula 2:

$$b = \sqrt{\left[\frac{((3*3)+(5*5))}{2}\right] - \left[\frac{(5-3)}{2}\right]}$$

$$b = 4 \text{ (B = b)}$$

Figure 3 shows the LNR processor based on the process, and its mathematical computational processor based on this process and condition. The LNR is a mathematical model that provides a linear relation between relations; thus, the LNR processor is completely based on this process and method without any modification to the formula. As we know, this paper introduces a conceptual logical LNR quantum computing processor, so this is a concept that provides an idea for the development of a processor that is currently implemented because of the expensive and error-handling of qubits. Quantum computing is very complex and computational, so it can be more complex to make design advances; therefore, only the current overview is introduced, but it is possible in the future. The LNR is a mathematical model that provides a linear relation between relations; thus, the LNR processor is completely based on this process and method without any modification to the formula.

Math Formulae

Formula 1

$$m = \sqrt{((b + b) - (b - b) - (b * b) - \left(\frac{b}{b}\right)) / (-1)}$$

$$n = \sqrt{((b + b) + (b - b) + (b * b) + \left(\frac{b}{b}\right)) / (1)}$$

Formula 2

$$b = \sqrt{\left[\frac{((a*a)+(c*c))}{2}\right] - \left[\frac{(c-a)}{2}\right]}$$

RESULT AND DISCUSSION

According to Figure 1, the LNR Quantum Processor is described, where B = 2, so M = 1 and N = 3, meaning B = B. This follows the Linear Number Relation (LNR) formula mentioned above. The formula operates using logical gates on qubits, which is simple in theory, but not necessarily easy to implement, as it involves complex conceptual and internal designs based on mathematical computations. Conceptually, the system works as intended, and this processor is designed specifically to produce the same number as the input value. It is a dedicated processor for the LNR formula and does not offer various features. However, its dedicated nature can provide benefits in complex applications, as its implementation reflects new avenues for evaluation and improvement.

CONCLUSION

The development of LNR quantum processors is a significant milestone in quantum computing research, particularly in the domain of linear number relations. This processor, designed by the author of this paper, represents a pioneering effort to conceptualize and implement a dedicated quantum computing system tailored specifically for evaluating linear number relation formulas.

The LNR quantum processor operates on sophisticated architecture, leveraging quantum gates and qubit arrangements to handle complex computations involving linear number relations. Through simulation and testing, the processor demonstrated its ability to efficiently evaluate linear number relation formulas, thereby providing accurate results with a high level of precision.

The testing process included inputting various datasets representing different linear number relation formulas, executing quantum circuits using the processor's quantum gates, and analyzing the measurement outcomes to validate the processor's functionality and performance. Performance metrics,

such as accuracy, speed, and resource utilization, are carefully monitored and evaluated to ensure that the optimal operation of the processor is based on theoretical assumptions.

Acknowledgment

I would like to express my sincere gratitude to all individuals and organizations who contributed to the development and realization of the LNR quantum processor. As an independent researcher, this project was the result of my dedication, passion, and perseverance in quantum computing. I also extend my appreciation to the quantum computing community for their valuable insights, discussions, and collaborations, which have enriched my understanding and contributed to the development of LNR quantum processors.

Furthermore, I acknowledge the guidance and inspiration drawn from various research papers, publications, and academic resources that have shaped my approach and methodology for designing quantum processors. Finally, throughout this study, I am thankful to my family and friends for their unwavering support, encouragement, and belief in my abilities.

REFERENCES

1. Barhate MN, Barhate NS, Barhate LN. Linear number relation. *Int J Novel Res Dev.* 2024;9:a374–a381.
2. Cacciapuoti AS, Caleffi M, Tafuri F, Cataliotti FS, Gherardini S, Bianchi G. Quantum internet: Networking challenges in distributed quantum computing. *IEEE Netw.* 2019;34:137–43. DOI: 10.1109/MNET.001.1900092.
3. Bravyi S, Gosset D, König R. Quantum advantage with shallow circuits. *Science.* 2018;362:308–11. DOI: 10.1126/science.aar3106. PubMed: 30337404.
4. Zhong HS, Wang H, Deng YH, Chen MC, Peng LC, Luo YH, et al. Quantum computational advantage using photons. *Science.* 2020;370:1460–3. DOI: 10.1126/science.abe8770. PubMed: 33273064.
5. Madsen LS, Laudenbach F, Askarani MF, Rortais F, Vincent T, Bulmer JFF, et al. Quantum computational advantage with a programmable photonic processor. *Nature.* 2022;606:75–81. DOI: 10.1038/s41586-022-04725-x. PubMed: 35650354.
6. Yuan X, Endo S, Zhao Q, Li Y, Benjamin SC. Theory of variational quantum simulation. *Quantum.* 2019;3:191. DOI: 10.22331/q-2019-10-07-191.
7. Eisert J, Hangleiter D, Walk N, Roth I, Markham D, Parekh R, et al. Quantum certification and benchmarking. *Nat Rev Phys.* 2020;2:382–90. DOI: 10.1038/s42254-020-0186-4.
8. Zizzi P. Quantum information hidden in quantum fields. *Quantum Rep.* 2020;2:459–88. DOI: 10.3390/quantum2030033.
9. Han YH, Cao C, Fan L, Zhang R. Scheme for implementing nonlocal high-fidelity quantum controlled-not gates on quantum-dot-confined electron spins using optical microcavities and photonic hyperentanglement. *Front Phys.* 2022;10:1006255. DOI: 10.3389/fphy.2022.1006255.
10. Xu F, Ma X, Zhang Q, Lo HK, Pan JW. Secure quantum key distribution with realistic devices. *Rev Mod Phys.* 2020;92:025002. DOI: 10.1103/RevModPhys.92.025002.